# Device-Free Daily Life (ADL) Recognition for Smart Home Healthcare using a low-cost (2D) Lidar

Zixiang Ma, John Bigham, Stefan Poslad, Bang Wu, Xiaoshuai Zhang, and Eliane Bodanese

School of Electronic Engineering and Computer Science

Queen Mary University of London

London, United Kingdom

Email: {zixiang.ma, John.bigham, stefan.poslad, bang.wu, xiaoshuai.zhang, eliane.bodanese}@qmul.ac.uk

***Abstract*—Device-free or off-body sensing methods such as Lidar can be used for location-related Activities during Daily Life (ADL) recognition without the need for the subject to carry less accurate on-body sensors and because some subjects may forget to carry them or maintain them to be operational (powered up), i.e., users can be device free and the method still works. Hence, this paper proposes an innovative method for recognizing daily activities using a state-of-art seq2seq Recurrent Neural Network (RNN) model to classify centimeter level accurate location data from a 360-degree rotating 2D Lidar device. We deployed and validated the system. The results indicate that our method can provide a centimeter-level localization accuracy of 88% when recognizing seventeen targeted location-related daily activities.**

***Keywords— Lidar; Smart Home; Healthcare; Location-based Services; Seq2seq; Activities during Daily Life (ADL); Human Activity (HAR); Recurrent Neural Network (RNN)***

## I. BACKGROUND AND RELATED WORK

The subject of this paper is the recognition of (human) activities during daily life (ADL) also referred to as Human Activity (HAR) in a home, solely based upon location tracking using 2D Lidar; no sensors are worn by the subject. Other sensors could be deployed to assist the recognition of ADLs, but here the focus is on what can be identified merely by location tracking. The target location space is a small house or flat where high location accuracy is necessary to discriminate between which landmarks in the accommodation the subject are nearby. Such small accommodation is of interest as it tends to be used for sheltered accommodation, e.g., for seniors or the elderly, those in rehabilitation and those recovering from severe depression. Today there are several indoor localization systems that support applications such as indoor navigation, proximity monitoring, and indoor emergency location systems. These have been developed because of the lack of the Global Navigation Satellite (Positioning) System (GNSS) signal indoors. The most popular indoor location Based Systems (LBS) is Wi-Fi, as it can make use of an already deployed Wireless Access Point (WAP) infrastructure to determine the position. Similarly, Bluetooth is another widely used wireless LBS technology that also operates in the 2.4 Gigahertz spectrum that is inexpensive and energy efficient. Bluetooth enabled receivers, such as smartphones can pick up these signals and respond accordingly when a Bluetooth beacon comes into range. They can also be deployed in an inverted manner, where the subject wears a beacon with, e.g., an accelerometer and magnetometer, and fixed devices in the home receive the beacon signal strength and other sensor data from the person. Both WiFi and Bluetooth estimate locations using forms of trilateration or by constructing a radio map prior to the location estimation and then matching the current signals to a set of locations in the radio map index by, e.g., the RSSI at the receivers. Inside a small home, trilateration does not provide good accuracy because of non-line of site issues. Radio maps mitigate these inaccuracies, but due to the variations in the signal strength for both WiFi and BLE, the estimates still limit the accuracy, typically to 1 to 2 m which is too inaccurate and typically needs to be 1 to 2 orders of magnitude lower to differentiate physical items such use sink, fridge or kettle in a small house. It is known that the use of WiFi and Bluetooth require the user to carry an on-body device. Some studies have shown that some target users such as seniors are more disinclined to use such eHealth wearables [1]. There is also the issue that such wearables require a degree of maintenance such as recharging batteries to keep them operational.

There has been much research investigating how to improve indoor LBS accuracy. For example, the work of [2], [3], [4], [5] demonstrates the benefits of using a hybrid approach. [2] deploys Bluetooth in the home using inverted RSSI fingerprinting, step counting, magnetometers and, importantly, focuses on detection of moving path segments (using dynamic time warping) or stationary waits or stays referred to as Stay Points and has achieved a high accuracy of recognition for elementary actions. Their work also suggests an activity-centric trajectory based prediction model is a practical solution for location estimation in homes that can be extended to design a suitable user-friendly Internet of Things (IoT) application. This is achieved by populating the radio map using sequential RSSI data collected along a pathway (e.g., couch to the dining area) or stationary positions (e.g., sleeping on the bed). The location is associated with ADL. However, such a multi-sensor hybrid LBS still suffers from the same limitations as the WiFi and Bluetooth methods in terms of insufficient location accuracy and require the use of wearables.

Another indoor LBS choice is UWB, which has a high accuracy, of a few centimeters. Recent reductions in price and size of the transceivers make deployment a good choice in the future. UWB devices use a very large bandwidth and can transmit high data rates over short distances at very low power levels. It is not affected by the existence of other communication devices or external noise [6], although, it can be affected by other wide spectrum devices if misconfigured, e.g., WiMAX and digital TV. [7]. Like Lidar, distance is based on time and for UWB location estimation is based on Time difference of arrival (TDOA)-based algorithms or Time of arrival or flight (TOA)-based algorithms. The use of narrow pulses makes UWB very tolerant to multipath effects. However, despite the accuracy, it does require the subject to wear a transmitter.

The final type of off-body sensor considered is light-based: various cameras or Light detection and ranging (Lidar), which uses pulsed laser light and measuring the TOA reflected pulses with a sensor. A major weakness of the use of cameras is that they are highly privacy-invasive. Until recently, Lidar devices were quite costly to purchase. There is also a range of Lidar devices that could be used for ADL such as flash Lidars that only face in a single direction (1D), line scanning sensors that swept a beam across a scene, taking measurements along a single (2D) plane and 3D Lidar [8]. N.B Both 2D and 3D Lidar device can be designed to rotate 360 degrees. The major application of Lidar is for mobile unmanned vehicles and robots to track objects around them as they move [8] and to support collisions avoidance. *To the best of our knowledge, no work has looked at using off-body low-cost (2D rather than 3D) Lidar devices to recognise ADLs.*

In addition, to selecting and deploying sensors to accurately measure object locations including people, we also need to consider how to analyse any associated activities `derived from this. Understanding the activities of a person inside a home requires contextual information related to their inherent surroundings to be recorded. One approach to modeling ADLs is based on a task-specific and intention-oriented plan representation language such as Asbru [9]. Its roots are in the modeling of medical protocols and monitoring the application of such protocols [10]. [11], [12], developed a recognition engine to detect ADLs that were modeled using ASBRU from sensor events, principally RFID tags. The engine generated a range of possible compliant ADL task sequences from a stream of sensor data to determine the ADL being conducted along with an assessment of their possibility. Another way of representing and modeling high tier behavior is workflows, such as using an augmented Petri Net [13]. However, workflows are too prescriptive in their ordering. If workflows are applied in dynamically changing environments, they require a large number of permutations to be explicitly enumerated. Workflows can scale poorly to cases where there are many possibilities, and this is often the case for goals performed by people [14]. In addition to scalability issues, it can be tough to manage the representation of priorities and ordering. Thus, more flexibility is required when modeling hierarchal ADLs. The Asbru language is a process representation language which has similarities to workflow modeling but has been designed to provide more flexibility than workflows. Asbru allows flexibility in how it can represent temporal events, namely their duration and sequence. ADLs have some attributes and characteristics which make them challenging to represent in a logical framework. These characteristics include the variable duration of the same task, variable ordering of the tasks, and the degree of overlap with other ADLs. Techniques which attempt to map these as a flat structure are problematic because they are unable to model flexible scenarios, such as interweaving ADLs. The ability to monitor interweaving ADLs is a crucial advantage [15].

However, why such a recognition process can be effective the process requires the enumeration of hierarchical plans for each ADL, i.e., a library of ADLs has to be created where each ADL is elaborated. The approach in this paper bypasses this process using an inductive approach where patterns in behavior are labeled. [16] has used a pattern mining approach where the emphasis was in detecting unexpected behaviors. *In this paper we, however, attempt to detect normal patterns using a recursive neural network and to depend on Lidar data as the only sensor data.*

The main contributions and novelty of this paper are 3-fold:

1) *We propose an accurate 2D Lidar-based wearable-free location determination method from which ADLs can be derived and be recognised.*

2) *We extend this method so that it can reduce invalid Lidar data collection and classify multiple users in real time.*

3) *We propose a state-of-art seq2seq model to analyse the Lidar data for ADLs without requiring the enumeration of hierarchical plans for each ADL to improve the ADL detection accuracy.*

The remainder of this paper is organized as follows: Section II introduces the methods we have adopted and describes the experiments we performed. Section III presents the performance of our seq2seq-based activity recognition method. Finally, Section IV concludes with a summary of the results of our experiments and future work.

## II. METHOD

Our system consists of two phases of research and development: the first phase is 2D Lidar-based indoor localization, which will offer localization results that will be used in recognition activity. The second phase is activity recognition that uses the processed location data as the inputs of our seq2seq model.

### *A. 2D Lidar Localization and Multi-user Tracking*

#### 1) Lidar Data Collection

A low cost, 2D, rotating Lidar system called RPLIDAR (version A1) from Slamtech[1] was used and configured to continuously scan a physical space until the user switches it off. For residential rooms such as a kitchen, there is often no user presence for much of the day. Hence, much of the recorded data contains no ADL leading to a massive, largely redundant data set. Hence, we use a threshold based on Hausdorff distance, which measures how far two subsets of a metric space are from each other representing two time sequential scans, to detect the presence of a moving user.

[1] Available from https://www.slamtec.com/en/Lidar/A1.

First, we convert the raw Lidar radial distance and angle data to Cartesian coordinates. The position of Lidar is denoted by the origin point (0, 0). Then the Hausdorff distance between different scans I and J is calculated. The Hausdorff distance between Lidar scans I and J is defined as:

$$H_d(I,J) = \max\{s_{i\in I} i_{j\in J} d(i,j), s_{i\in I} i_{j\in J} d(i,j)\} \qquad (1)$$

where s represents the supremum, and i represents infimum. d(x, y) means the Euclidean distance between a point i in the scan I and point j in scan J.

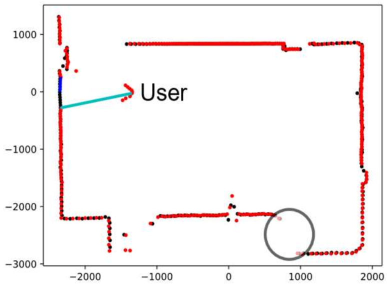
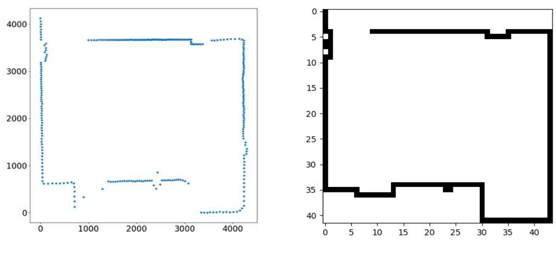



**Figure 1.** How the Hausdorff distance works.

The origin point (0,0) is where the Lidar is located, and the distance measuring unit is millimeters. The black points are from a scan (e.g., scan 1), without any moving objects. The red points are from a scan (scan 27) where a user walks into the kitchen. The length of the duck egg blue line shown in figure 1 means the $sup_{x\in X} inf_{y\in Y} d(scan27, scan1)$, which is also the directed (or forward) Hausdorff distance between scan 27 and scan 1. The distance of the blue line equals to $sup_{x\in X} inf_{y\in Y} d(scan1, scan27)$. User movement is what causes the blue line. When a user moves the kitchen boundary beyond the user from the Lidar device is not detected as the user blocks the light beam.

Instead of using Hausdorff distance, we also tried to differentiate between two scans based on comparing distances at each angle. However, a small angle difference can lead to a considerable difference in distance, which is shown in Figure 1 (the black circle).

2) 2D Lidar Data Pre-processing



(a) (b)

**Figure 2.** Building a grid map from Lidar points data where Each grid equals 10*10 cm$^2$.

The raw data was collected from the 2D Lidar chosen consists of time, distance, angle, and quality. Because of the noisy and uncertain sensor measurements, we use Occupancy Grid Mapping (OGM) [17], an algorithm often adopted by mobile robots, to address the uncertain data problem, and to build a 'stable' floor plan.

3) Density-based spatial clustering of applications with noise

Because of the system error when using a 2D Lidar device, a few outliers will also show in our processed location-based 'image map.' This would severely reduce the localization and recognition accuracy, especially when there are more than two users in the same room. So, we use Density-based spatial clustering of applications with noise (DBSCAN) to reduce anomalies and represent different users as clusters.

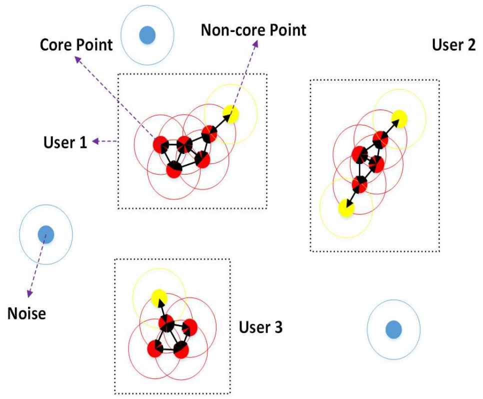



**Figure 3.** Three clusters are generated (for minPts=2).

DBSCAN is a clustering algorithm based on regional data density, which is often used for outlier detection. It also marks outliers lying in low-density regions for removal. The algorithm requires two parameters; one is a radius ε, another one is minPts. Points are classified as follows:

- A core point means at least minPts points are within a ε radius. Each cluster consists of at least one core point.
- A non-core point means the point we can directly reach is less than minPts core points. We can count the non-core points into the same cluster, or delete them – it does not matter. They cannot be used to reach more points.
- All points that are not reachable from any other point are considered as noise points (outliers).

## *B. Human Daily Activities Recognition*

1) Recurrent Neural Networks

Our target is to recognise location-related human daily activities, the inputs are sequential location data, in our case, 2D Lidar data. A natural choice to classify sequential data is to use a Recurrent Neural Network (RNN). We selected Long short-term memory (LSTM)[2] as our baseline algorithm to recognise activities, which is one of the most used RNNs.

Then the research question for our project is how to build the corresponding output activities. Since the kitchen has various landmarks, such as a table, sink, fridge, and

[2] We used Keras an open source neural network library written in Python to implement the LSTM that is easy to use. It's available from https://keras.io/.

microwave, Lidar can be used to track motion between landmarks and the continued human presence at a landmark – a so-called stay point. One solution to define those output activities can be based on those landmarks, as most activities happen near those landmarks. However, there could be a lot and different landmarks in different user's house. So we use a Stay Point Recognition algorithm to recognise those key landmarks.

2) Stay Point Recognition

We first segment the data into fixed time periods of 30 seconds and then to identify points where the subject stays for at least a few seconds, so called stay points. 30s was chosen as a trade-off between allowing sufficient time to link and analyse a small chain of landmarks visits that may be stay points, together as simple ADLs versus creating and having to analyse much more complex ADL chains. We use a stay point recognition algorithm [18] to identify key landmarks in the time segment, as our basic assumption is that staying at a location means that this point is significant. The subject may well pass other landmarks, but these are not necessarily relevant to the activity being performed.

The basic idea for stay point recognition is as follows:

1. The trajectory is represented as the set:

$$S = \{loc_k = (t_k, l_k) | k = 1,2..n\} \quad (1)$$

where l is the cartesian coordinates, t is the timestamp.

2. We transform the data into the following equations:

$$f(t_k) = t_k - t_e \quad (2)$$

$$Z = g(f(t_k)) = \sum_{n=2}^{k} distance(l_n, l_{n-1}) \quad (3)$$

$t_e$ is the time when the subject enters the kitchen and distance is the Euclidean distance. The horizontal axis in figure 4 represents $f(t_k)$ is the time when the subject enters the kitchen and $distance(l_k, l_{k-1})$ is the Euclidean distance between two points. The horizonal axis in figure 4 represents $f(t_k)$ and the vertical axis represents Z, which is a monatomic increasing curve. In equation (3), we subtract the time of entry because no data collection is logged when the kitchen is empty.

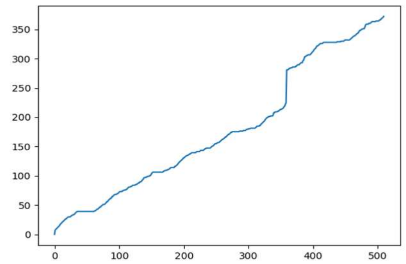


**Figure 4.** After transformation to function Z.

As $g(l_k)$ increases, its first derivative will be always greater or equal to zero. The stay points should be where g is horizontal, i.e., where the local minima of the derivative of the curve. To identify the local minima, we check the zero-crossing of the second derivative, as the second derivative may not equal to zero as g is a discrete function. Figure 5 depicts the second derivative, and the red boxes means the selected regions.

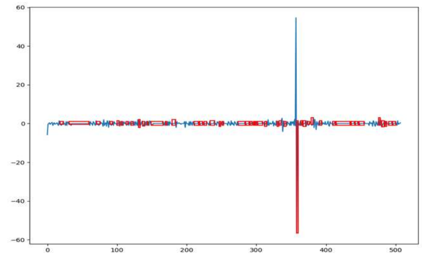


**Figure 5.** The second derivative of the curve.

The red boxes not only include the stay points but also include inflections, which are caused when you walk slowly, then walk speedily.

To reduce the inflections, we also use a confidence value to recognise the stay point. Based on paper [18], we also define a single confidence value of data point as:

$$C(P_i) = 100 \text{ g'} \leq 0.01 \quad (4)$$

$$\text{Else: } C(P_i) = 100 - \frac{g' - 0.01}{0.01} \quad (5)$$

$$C(P_k, P_w) = \frac{\sum_{i=k}^{w} C(P_i)}{w-k+1} \quad (6)$$

where g' is the first derivative. We set $C(P_k, P_w)$ in $C(P_i, P_j)$. i, j are the boundary points of zero-crossing of region k. w is where the confidence level of this sub-region is above 80. The center of those points will be treated as a stay point.

3) Sequence-to-sequence Model

Seq2seq is a general-purpose encoder-decoder model, which is initially built for machine translation [19], but that has also been used for a wide variety of tasks, e.g., image captioning and text summarization.

The reason that we chose to use a seq2seq model is that we thought the sequential location information could be analogous to a translation problem. Each sequential location information belongs to different corresponding activities.

Because activities can take different amounts of time, selecting fixed periods, such as 30 seconds, and mapping each 30 second period to one activity does not work. There can, for example, be more than one activity in 30 seconds. Hence the use of a vanilla neural network also does not work. To collect the training data, we segmented the 30 seconds using stay points and then annotate the stay points and annotate sub-sections of each segment with the elementary activities. Lidar data was collected in a kitchen. We asked two users to label the stay points and related activities. Then the trajectory of every 30s will be the input data to train the seq2seq2 model. The corresponding activity will be the labeled data, e.g., in an example 30s time slot, the corresponding activities are {from table to fridge, use fridge, prepare to cook}, then those activities can be represented as {2, 3, 8} (see the list given in Section III), then based on the activity, identifier we converted these into a one-hot encoding, where categorical variables are converted into a 0/1 form, which then forms the input into the seq2seq model. After this transformation, more sequential training data will be generated to train our model.

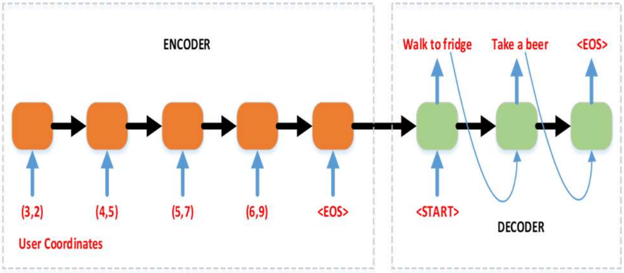



**Figure 6.** The sequential coordinates will be the input of the encoder; then the decoder will output the corresponding activities, where <EOS> means the end of the sequence.

## III. Evaluation

### *A. Experimental Setup*

The dataset we collected and analysed and consisted of a total of 536 times 30s time slots of acquired Lidar data, collected for 2 users. Of this, we used 100 times 30s for the test set and the rest for the training set. The experiment area is a kitchen as shown in Figure 2 and covers about 20 $m^2$. The Lidar we used is a 2D RPLIDAR, which is shown in figure 7.

**Figure 7.** The 2D RPLIDAR device.

It supports a 5-10Hz Adaptive Scan Frequency, 0.2 to10 m (in our test) distance range, and is low cost compared to older rotating 2D Lidar devices (currently it costs about £ 120).

### *B. Human Activity Recognition Performance*

For the localization part, the system works well. Based on the comparison, each data point can be classified to a real grid. However, if two users are close enough (less than 20 cm apart), two clusters may merge into one, but this was not a big issue. A user that is detected by the system could become undetected within a later period in the same sequence, that did not leave via a door stay point. We could then perhaps assume the user has fallen, as the Lidar scan is set to a certain height (1.1 m in our case). 1.1 m is chosen as a trade-off between being low enough to detect kitchen units that are offset from the wall so we can detect we are at them versus being high enough to detect an upright human. If a user becomes undetected, the user must be at a height less than 1.1m, which we could presume the user has fallen down.

For human activity recognition, we use the stay point as the reduced set of important landmarks and as an activity connector. So, the first step is that we recognise the user's stay points, then ask the user to label the stay points and the activity category. The next step is to use the sequential location data as inputs to train the model, where the corresponding label data represents the labeled activities (see figure 6).

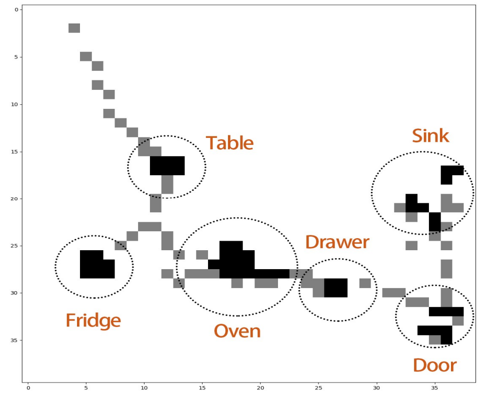



**Figure 8.** Labelled stay points

In Figure 8, the grey grids represent a user's trajectory over a period. The black ones mean the recognised stay points. To better visualise the result, we only visualise part of the dataset. Here, six stay points (landmarks) are identified. We asked the user to label those stay points and to point out the related activities. There are several ways to label ADLS. First, we set several elementary ADLs, i.e., eating, drinking or use the fridge, then ask the user to label the corresponding sequential location data. Another way is to ask the user to label personalized activities. In our experiment, there are 17 activities labeled by the user based on daily simple kitchen activities. These are: 1) from door to table, 2) from table to fridge, 3) use fridge, 4) fridge to food pantry, 5) prepare cat food, 6) food pantry to back door, 7) feed the cat, 8) no one in the kitchen, 9) open back door, 10) back door to fridge, 11) prepare to cook, 12) oven to fridge, 13) fridge to back door, 14) use breadboard (drawer), 15) use oven, 16) go out of back door, 17) washup.

One thing which should be noticed is that the labeled activities can include several different hierarchical levels. For example, we have recognised the 'food pantry' as a stay point, in a specific 30s, the user can either act walking from the food pantry to backdoor but or by staying at or very near to the food pantry. A user can also label their actions at another level. A user is not only near a stay point; the user is doing ADLs associated with a stay point such as preparing cat food in the food pantry or putting something into the food pantry, which is also like a text translation problem, the terms can have a different meaning.

Although the sequence recognition accuracy is reasonable at 88%, when the predicted sequences are the same as the input ones, a few activities from the total activity sequence are wrongly recognised, which is shown in figure 9. The darker blue, the more accurate the recognition. 0 to 16 represent the 17 labeled activities.

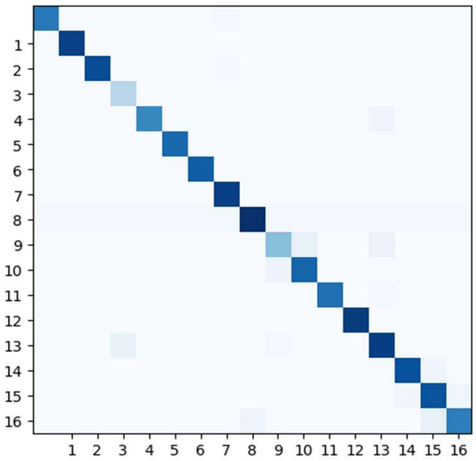



**Figure 9.** Recognition confusion matrix

## IV. Conclusion

Traditional ADL usually requires users to carry on body sensors to generate the data that is analysed via an enumeration of hierarchical plans. In contrast, we proposed, developed and validated 2D Lidar measurements of ADL (in a kitchen area used by two different people at different times). We combined the use of Lidar with the state-of-art seq2seq RNN model to classify ADLs linked to the Lidar generated stay points and transitions between stay points. Our validation shows that a 2D Lidar location determination method can provide centimeter-level localization accuracy and a good accuracy (88%) in recognizing seventeen location-related daily activities.

Our future research extensions are as follows. First, instead of using sequential location inputs, we will investigate the use of sequential landmarks that are stay points as the training data, which may make the system more adaptive to different scenarios. Second, we will extend the evaluation of our system in different ADL scenarios that involve more users. Third we aim to investigate the application of other state-of-art zero/one-shot learning to use less samples to train the system. Fourth we will combine our previous research: using RSSI localization methods [3] to lift the accuracy of the recognition result. Fifth, we will look at more distributed AI algorithms to analyse ADL data [20], [21]. Finally, we will investigate Lidar use for robotic activity, including its use to recognise combined human and robot activities in a shared space as there is little work on the use of 2D Lidar for activity tracking by humans and robots.

## Acknowledgments

This research was funded in part by a China Scholarship Council (CSC) and QMUL PhD Grant. It is also funded in part by and by the UK EPSRC National Centre for Nuclear Robotics, NCNR project (EP/R02572X/1). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for the data analytics.

## References

[1] D. M. Levine, S. R. Lipsitz, and J. A. Linder, “Trends in seniors use of digital health technology in the united states, 2011-2014,” Jama, 2016, vol. 316, no. 5, pp. 538–540.

[2] M. Sridharan, J. Bigham, C. Phillips, and E. Bodanese, “Collaborative location estimation for confined spaces using magnetic field and inverse beacon positioning,” in SENSORS, 2017 IEEE. IEEE, 2017, pp. 1–3.

[3] Z. Ma, S. Poslad, S. Hu, and X. Zhang, “A fast path matching algorithm for indoor positioning systems using magnetic field measurements,” in Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on. IEEE, 2017, pp. 1–5.

[4] Z. Ma, S. Poslad, J. Bigham, X. Zhang, and L. Men, “A ble rssi ranking based indoor positioning system for generic smartphones,” in Wireless Telecommunications Symposium (WTS), 2017, pp. 1–8.

[5] K. A. Nguyen, C. Watkins, and Z. Luo, “Co-location epidemic tracking on london public transports using low power mobile magnetometer,” arXiv preprint arXiv:1704.00148, 2017.

[6] S. Krishnan, P. Sharma, and O. H. Woon, “A uwb based localization system for indoor robot navigation,” in IEEE Int. Conf. on.Ultra-Wideband, 2007. ICUWB 2007, pp. 77–82.

[7] R. Aiello and A. Batra, Ultra wideband systems: technologies and applications. Newnes, 2006.

[8] J. Shackleton, B. and VanVoorst, “Tracking people with a 360-degree lidar,” in Advanced Video and Signal Based Surveillance (AVSS), Seventh IEEE International Conference on. IEEE, 2010, pp. 420–426.

[9] C. Fuchsberger, J. Hunter, and P. McCue, “Testing asbru guidelines and protocols for neonatal intensive care,” in Conference on Artificial Intelligence in Medicine in Europe. Springer, 2005, pp. 101–110.

[10] R. Kosara, S. Miksch, Y. Shahar, and P. Johnson, “Asbruview: capturing complex, time-oriented plansłbeyond flow charts,” in Diagrammatic Representation and Reasoning. Springer, 2002, pp. 535–549.

[11] U. Naeem and J. Bigham, “A comparison of two hidden markov approaches to task identification in the home environment,” in Pervasive Computing and Applications, 2007. ICPCA 2007. 2nd International Conference on. IEEE, 2007, pp. 383–388.

[12] U. Naeem, J. Bigham, and J. Wang, “Recognising activities of daily life using hierarchical plans,” in European Conference on Smart Sensing and Context. Springer, 2007, pp. 175–189.

[13] R. Zurawski, “Petri nets and industrial applications: A tutorial,” IEEE Transactions on industrial electronics, vol. 41, no. 6, pp. 567–583, 1994.

[14] W. van der Aalst, “How to handle dynamic change and capture management information?” in 4th International Conference on Cooperative Information Systems, Edinburgh, Scotland. Citeseer.

[15] S. Nasreen, M. A. Azam, U. Naeem, M. A. Ghazanfar, and A. Khalid, “Recognition framework for inferring activities of daily living based on pattern mining,” Arabian Journal for Science and Engineering, vol. 41, no. 8, pp. 3113–3126, 2016.

[16] U. Naeem and J. Bigham, “Activity recognition in the home using a hierarchal framework with object usage data,” Journal of Ambient Intelligence and Smart Environments, vol. 1, no. 4, pp. 335–350, 2009.

[17] T. Colleens and J. Colleens, “Occupancy grid mapping: An empirical evaluation,” in IEEE Mediterranean Conferece on Control & Automation, 2007, pp. 1–6.

[18] G. Stylianou, “Stay-point identification as curve extrema,” arXiv preprint arXiv:1701.06276, 2017.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Advances in neural information processing systems, 2014, pp. 3104–3112.

[20] S. Poslad, P. Buckle and R.G. Hadingham “FIPA-OS: the FIPA agent Platform available as Open Source”, 5th Int. Conf on the Practical Application Of Intelligent Agents And Multi-Agent Technology (PAAM) 2000, pp. 355-368.

[21] Poslad S, Buckle P, Hadingham R.G. Open Source, Standards and Scaleable Agencies. Lecture Notes in Computer Science, 2001, vol. 1887, pp. 296-303.